EL970861029US

DATABASE STRUCTURE AND FRONT END

Inventors: Dominic Bennett and Dan Hu

BACKGROUND OF THE INVENTION

1. Field Of The Invention

The present invention relates generally to databases, and more particularly, but not exclusively, to database structure and front ends.

2. Description Of The Background Art

Large public computer networks, such as the Internet, allow advertisers to reach a worldwide audience twenty-four hours a day, seven days a week. This has made large public networks a cost-effective medium for marketing and selling products (e.g., goods and services). On the Internet, for example, advertising revenues allow companies to distribute free software or provide free access to websites. Needless to say, advertising helps fuel the Internet economy.

In order to provide relevant advertisements to consumers, companies engaged in online advertising maintain databases of advertising-related data. Such databases need to be accessed by sales and marketing personnel as they are the ones who typically plan and implement advertising campaigns. Unfortunately, some sales and marketing personnel are non-technical, and thus have difficulty working with the database. Front ends, which are application programs for interfacing with databases, may be provided to assist non-technical users in accessing the database. However,

conventional front ends get harder to use as the number of selection choices for the database increases.

A database for storing online advertising-related data can grow very quickly because of the large number of consumers on the Internet. If a database is not structured properly, accessing the database may take longer as more data are stored in it. As a result, reports generated from the database may also take longer. This may discourage sales and marketing personnel from generating reports, and may keep some reports from being generated on time.

From the foregoing, an improved database structure and front end are generally desirable.

SUMMARY

In one embodiment, a method of analyzing online advertising information includes the steps of receiving consumer data from client computers, creating a database based on the consumer data, receiving user selected values from a front end, and extracting data from the database based on the user selected values. The front end may have a selection area with user selectable values that change depending on an initially selected value. In one embodiment, the database comprises an online analytical processing (OLAP) database.

These and other features of the present invention will be readily apparent to persons of ordinary skill in the art upon reading the entirety of this disclosure, which includes the accompanying drawings and claims.

DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a schematic diagram of an example computer that may be used in embodiments of the present invention.

FIG. 2 shows a schematic diagram of a computing environment in accordance with an embodiment of the present invention.

FIG. 3 shows a schematic diagram of an OLAP database and a front end program in accordance with an embodiment of the present invention.

FIG. 4 shows a screenshot of an example layout screen in accordance with an embodiment of the present invention.

FIG. 5 shows a screenshot of an example filter screen in accordance with an embodiment of the present invention.

FIG. 6 shows a screenshot of an example report displayed in a report screen in accordance with an embodiment of the present invention.

FIG. 7 shows a screenshot of an example screen for a scheduler.

FIG. 8 shows a screenshot of an example screen for an alerts.

FIG. 9 shows a flow diagram of a method of generating a report in accordance with an embodiment of the present invention.

The use of the same reference label in different drawings indicates the same or like components.

DETAILED DESCRIPTION

In the present disclosure, numerous specific details are provided such as examples of apparatus, components, and methods to provide a thorough understanding

of embodiments of the invention. Persons of ordinary skill in the art will recognize, however, that the invention can be practiced without one or more of the specific details. In other instances, well-known details are not shown or described to avoid obscuring aspects of the invention.

Being computer-related, it can be appreciated that the components disclosed herein may be implemented in hardware, software, or a combination of hardware and software (e.g., firmware). Software components may be in the form of computer-readable program code stored in a computer-readable storage medium, such as memory, mass storage device, or removable storage device. For example, a computer-readable storage medium may comprise computer-readable program code for performing the function of a particular component. Likewise, computer memory may be configured to include one or more components, which may then be executed by a processor. Components may be implemented separately in multiple modules or together in a single module.

Embodiments of the present invention are described herein in the context of advertising delivery over the Internet. It should be understood, however, that embodiments of the present invention may be generally employed to build databases and front ends for databases.

Embodiments of the present invention employ a message delivery program in communication with a message server. Message delivery programs and message servers are also disclosed in the following commonly-assigned disclosures, which are incorporated herein by reference in their entirety: U.S. Application No. 10/152,204, filed on May 21, 2002, and U.S. Application No. 10/289,123, filed on November 5, 2002.

Referring now to FIG. 1, there is shown a schematic diagram of an example computer that may be used in embodiments of the present invention. Depending on its configuration, the computer shown in the example of FIG. 1 may be employed as a client computer or a server computer. The computer of FIG. 1 may have less or more components to meet the needs of a particular application. As shown in FIG. 1, the computer may include a processor 101, such as those from the Intel Corporation or Advanced Micro Devices, for example. The computer may have one or more buses 103 coupling its various components. The computer may include one ore more input devices 102 (e.g., keyboard, mouse), a computer-readable storage medium (CRSM) 105 (e.g., floppy disk, CD-ROM), a CRSM reader 104 (e.g., floppy drive, CD-ROM drive), a display monitor 109 (e.g., cathode ray tube, flat panel display), a communications interface 106 (e.g., network adapter, modem) for coupling to a network, one or more data storage devices 107 (e.g., hard disk drive, optical drive, FLASH memory), and a main memory 108 (e.g., RAM). Software embodiments may be stored in a computer-readable storage medium 105 for reading into a data storage device 107 or main memory 108. In the example of FIG. 1, main memory 108 may be configured to include a front end program 220, which is further discussed below. A front end program 220 may be executed by processor 101.

FIG. 2 shows a schematic diagram of a computing environment in accordance with an embodiment of the present invention. In the example of FIG. 2, websites 112 (i.e., 112-1, 112-2,...) comprise web servers accessible over the Internet. A website 112 may provide news, search engines, forums, audio and video streaming, e-mail service, and so on. A website 112 may provide information by way of web pages.

A client computer 130 may comprise a computer operated by a consumer navigating on the Internet. A client computer 130 may comprise a personal computer running the Microsoft Windows™ operating system, for example. Depending on the application, a client computer 130 may also be a portable or hand-held device, such as a laptop computer, a personal digital assistant, a digital mobile telephone, and so on. A client computer 130 may include a web browser 132 to allow a consumer to view web pages on websites on the Internet. A web browser 132 may be a commercially available web browser, such as the Microsoft Internet Explorer™ web browser. A web browser 132 allows a client computer 130 to receive one or more web pages from among available websites 112.

A client computer 130 may include a message delivery program 160. A message delivery program 160 may initiate the displaying of a presentation vehicle 162 to display an advertisement 166. Presentation vehicle 162 may be a browser window or a custom window. For example, presentation vehicle 162 may be a pop-up or a pop-under window. In one embodiment, a message delivery program 160 is downloadable from a message server computer 163.

A message delivery program 160 may be downloaded in conjunction with the downloading of another computer program. For example, a message delivery program 160 may be downloaded to a client computer 130 along with a utility program 167 that is provided to the consumer free of charge or at a reduced cost. A utility program 167 may be an e-wallet or an appointment calendar, for example. A utility program 167 may be provided to a consumer in exchange for the right to deliver advertisements to the consumer via a message delivery program 160. In essence, revenue from

advertisements delivered to the consumer helps defray the cost of creating and maintaining the utility program 167.

In one embodiment, a message delivery program 160 is a client-side program that monitors the online activity of a consumer across several websites, and reports its observations to a message server 163. It is to be noted that the mechanics of monitoring a consumer's online activity, such as determining where a consumer is navigating to, what a consumer is typing on a web page, when a consumer activates a mouse or a keyboard, when a consumer clicks on an advertisement, and the like, is, in general, known in the art and not further described here. For example, a message delivery program 160 may listen for event notifications from a web browser 132 as part of its monitoring function. A message delivery program 160 may protect the consumer's privacy by maintaining the consumer's anonymity (e.g., by using a machine ID to refer to the consumer) and encrypting sensitive information, such as credit card numbers.

In one embodiment, a message delivery program 160 monitors a web browser 132 for the uniform resource locator (URL) of websites visited by the consumer. A message delivery program 160 also keeps track of the number of impressions (i.e., displaying) of an advertisement 166 in the client computer 130, as well as the number of times the consumer clicked on an advertisement 166. A message delivery program 160 may periodically provide a data packet 168 containing its observations to a message server 163. Data provided by a message delivery program 160 to a message server 163 are also referred to as "consumer data." Consumer data thus include information on the websites visited by a consumer, URLs of web pages viewed by the consumer,

the number of impressions of advertisements in the consumer's client computer, and the number of times the consumer clicked on advertisements.

A client-side program, such as a message delivery program 160, allows for web-wide monitoring of consumer online activities. Unlike a website, which can only monitor consumer behavior on the website or related websites, a message delivery program 160 can advantageously collect consumer data across multiple, un-related websites. Consumer data collected by a message server 163 from a large number of message delivery programs 160 are thus good indicators of consumer need, as well as the effectiveness of an advertising campaign.

A message server 163 may comprise a server computer in communication with a message delivery program 160. Note that a message server 163 typically works in conjunction with a plurality of client computers 130, each having a message delivery program 160; only one client computer 130 is shown in FIG. 2 for clarity of illustration. Message server 163 may include a data warehouse 171 for storing consumer data received from client computers 130. Data warehouse 171 may be a commercially available database, such as those of the type available from the Oracle Corporation of Redwood Shores, California. In one embodiment, a message server 163 includes an online analytical processing (OLAP) database 174, which may also be of the type available from the Oracle Corporation. An OLAP database 174 contains a subset of consumer data from a data warehouse 171, as well as advertising data, such as advertiser names, contracts with advertisers, advertising campaigns, and so on. As will be more apparent below, an OLAP database 174 may be structured to allow for the use of hierarchical tables that better organize advertising data and facilitate data access.

Still referring to FIG. 2, a message server 163 may include a procedure 175. In one embodiment, a procedure 175 comprises computer-readable program code for receiving dimensions and facts from a front end program 220, querying an OLAP database 174 based on the received dimensions and facts, filtering the result of the query based on the received dimensions, and providing the filtered result to the front end program 220. The terms "dimensions" and "facts," which are used herein in the context of an OLAP database cube, are further discussed below.

A client computer 210 may be in communication with a message server 163. In one embodiment, a client computer 210 comprises a personal computer running the Microsoft Windows™ operating system. A client computer 210 may include a front end program 220. A front end program 220 may comprise computer-readable program code for accepting dimensions and facts from a user, providing the dimensions and facts to a procedure 175, receiving a filtered result from the procedure 175, and presenting a corresponding report to the user. A front end program 220 may communicate with a procedure 175 using client-server protocol. A client computer 210 may also include productivity programs 222, such as the Microsoft Excel™ spreadsheet, Microsoft Power Point™ presentation program, Microsoft Word™ word processing program. A front end program 220 may be employed in conjunction with productivity programs 222 to display and analyze reports that are based on data extracted by a procedure 175 from an OLAP database 174.

Turning now to FIG. 3, there is shown a schematic diagram of an OLAP database 174 and a front end program 220 in accordance with an embodiment of the present invention. As shown in FIG. 3, an OLAP database 174 may comprise hierarchy

tables 340 (i.e., 340-1, 340-2,...), a dimensions control table 342, and a hierarchy control table 344. Hierarchy tables 340 may comprise a database table configured to have dimensions and facts, with each fact being associated with a single dimension or combination of dimensions. Hierarchy tables 340 are arranged in a hierarchical topology, with the lowest level hierarchy table 340 having the most number of dimensions, the next higher level hierarchy table 340 having less dimensions than the lowest hierarchy table 340, the next next higher level hierarchy table 340 having less dimensions than the next hierarchy table 340, and so on. That is, the hierarchy tables 340 may be configured as follows:

(1) First level (lowest level) hierarchy table:

$Dimension_1$, $Dimension_2$,....$Dimension_n$; $Fact_1$, $Fact_2$,...$Fact_k$.

(2) Second level hierarchy table:

$Dimension_1$, $Dimension_2$,....$Dimension_{n-1}$; $Fact_1$, $Fact_2$,...$Fact_k$.

(3) Third level hierarchy table:

$Dimension_1$, $Dimension_2$,....$Dimension_{n-2}$; $Fact_1$, $Fact_2$,...$Fact_k$

and so on. Note that a second level hierarchy table may have one less dimension than the first level hierarchy table, while a third level hierarchy table may have one less dimension than the second level hierarchy table. Further note that the number of facts in each hierarchy table does not necessarily have to be different.

The hierarchical levels allow for faster data access in a level "n" compared to a level "n-1". That is, data can be accessed faster in the higher levels. The idea is to

minimize the number of rows by eliminating dimensions successively. The elimination of dimensions results in smaller tables, which results in faster data access.

In one embodiment, the dimensions and facts in hierarchy tables 340 relate to online advertising. The dimensions may include advertisements, campaigns, contracts, and other advertising data. As a further example, each advertisement may belong to one or more advertising campaigns, with each advertising campaign being associated with one or more contracts, and so on. The facts may include impressions (i.e., displaying of an advertisement) and clicks on impressions. Thus, for each particular value of advertisement, campaigns, contracts, or combinations thereof there may be a corresponding impressions value and clicks value. For example, a particular advertisement for a particular advertising campaign may have 2,000 impressions. As another example, a particular advertising campaign with an associated contract may have resulted in 4,000 impressions and 1,000 clicks on the impressions. Of course, the number and type of dimensions and facts, and their corresponding values, may vary to meet the needs of specific applications.

Each hierarchy table 340 may be structured to have the facts for a particular combination of dimensions. As a particular example, assuming an exhaustive list of dimensions consists of advertisement, advertising campaign, and contract, a first (lowest) level hierarchy table 340 will include facts (e.g., impressions, clicks, or both) for the dimensions advertisement, advertising campaign, contract, or combinations thereof; a second level hierarchy table 340 will include facts for the dimensions advertisement, advertising campaign, or combinations thereof; and so on. That is, for a particular number of dimensions, there will be a hierarchy table 340 with the corresponding facts.

In one embodiment, hierarchy tables 340 are manually populated using data from a data warehouse 171. Hierarchy tables 340 may also be populated using a script, for example. As can be appreciated, extracting data from a hierarchy table is generally faster than extracting data from an entire database. Hierarchy tables 340 thus provide a database structure that advantageously allows for relatively fast data access.

We need to communicate that the front end is "thought" about the exhaustive list of all dimensions and facts, the relationship between each dimension (i.e. the hierarchy), the table names of each level of aggregation and the dimensions available in each aggregate table, the type of each dimension i.e. free form, tree or list box by using control tables.

In one embodiment, a dimension in an OLAP database 174 may be one of three kinds of dimensions namely, "pull-down," "tree," or "free-form." A pull-down dimension may have a value that is selectable from a pull-down menu. For example, assuming "advertisement" is a pull-down dimension, the values "Ad1" for a first advertisement or "Ad2" for a second advertisement may be selected in a pull-down menu for "advertisement." A tree dimension may have a value that is selectable from a hierarchical tree structure. For example, assuming "category" is a tree dimension, the values "automotive" for web pages relating to automotives or "travel" for travel-related web pages may be selected in a tree structure for "category." In the tree structure, the value "hotel" may be included as branching off the value "travel." A free-form dimension may have a value that may entered without choosing from available selections. For example, assuming "revenue" is a free-form dimension, a user may enter any revenue amount for "revenue."

As shown in FIG. 3, an OLAP database 174 may include a dimensions control table 342. A dimensions control table 342 may comprise an exhaustive list of all dimensions and facts in all hierarchy tables 340, templates for constructing queries for each dimension, and a dimension look-up table. The dimension look-up table is employed in embodiments where the hierarchy tables 340 refer to dimensions using identifiers other than the dimensions' actual names (e.g., using "d2345" in a hierarchy table 340 to refer to a dimension "advertisement"). The dimension look-up table allows for translation of an identifier to actual name, and vice versa. A dimensions control table 342 may also have information on the kind of each dimension (e.g., whether a dimension is a pull-down, tree, or free-form) and conditional operators that may be used for a particular dimension.

An OLAP database 174 may also include a hierarchy control table 344. A hierarchy control table 344 may comprise information indicative of the data structure of the OLAP database 174. In one embodiment, a hierarchy control table 344 identifies each hierarchy table 340, the hierarchical order of the hierarchy tables 340, and the dimensions included in each hierarchy table 340. A hierarchy control table 344 may thus be consulted to identify the highest level hierarchy table 340 containing a particular set of dimensions and filters. Note that the highest level hierarchy table containing a particular set of dimensions and filters would advantageously have the least number of rows among hierarchy tables that also contain the set of dimensions and filters.

The general role of the front end is to optimally obtain that slice of the OLAP cube that contains all the data that the user is interested in viewing and displaying that slice of the OLAP cube in exactly the format that the user wishes to view. The filter page

allows the user to slice the OLAP cube using criteria on the dimensions. The layout page allows the user to specify the format of the report.

Still referring to FIG. 3, a front end program 220 may comprise a user interface 360, a scheduler 374, and an alerts 376. In one embodiment, a front end program 220 is implemented using the Microsoft Visual Basic For Applications™ (VBA) programming language. A user interface 360 may comprise computer-readable program code for allowing a user to enter selection criteria and generate a report in accordance with the selection criteria. The selection criteria may be values for dimensions and facts. A user interface 360 may comprise a layout screen 362, a filter screen 363, and a report screen 364.

A layout screen 362 allows a user to select dimensions and facts of interest. The selected dimensions and facts will be the basis of a subsequently run report. Generally speaking, a layout screen 362 allows the user to specify the format of the report. FIG. 4 shows a screenshot of an example layout screen 362 in accordance with an embodiment of the present invention. In the example of FIG. 4, an exhaustive list of all available dimensions and facts in hierarchy tables 340 is displayed in a window 410. The user may select one or more available dimensions and facts from the window 410 for inclusion in selection areas 402, 404, 406, and 408. In the example of FIG. 4, selection areas 402, 406, and 408 only accept one or more dimensions, while selection area 404 only accepts facts. User interface 360 enforces the rules on which item in window 410 can be placed in which selection area (i.e., dimensions can only go to selection areas 402, 406, and 408; facts can only go to selection area 404). In accordance with standard OLAP terminology, dimensions in selection area 402 are also

referred to as "column edge dimensions," dimensions in selection area 406 are also referred to as "row edge dimensions," and dimensions in selection area 408 are also referred to as "page edge dimensions." In the example of FIG. 4, the user has selected "ByDay" as a column edge dimension, "campaign" as a row edge dimension, "advertiser" as a page edge dimension, and "impressions," clicks", and "CTR" as facts. "CTR" stands for click-through-rate, and is a calculated fact obtained by dividing the number of impressions with the number of clicks. Click-through-rate is a measure of the effectiveness of an advertisement. User interface 360 may also employ visual cues to assist users in working with items in window 410.

A filter screen 363 accepts dimensions that will be used as filters to the dimensions and facts selected in a layout screen 362. The general role of a front end program 220 is to optimally obtain that slice of an OLAP cube that contains all the data that the user is interested in viewing, and displaying that slice of the OLAP cube in exactly the format that the user wishes to view. In that regard, a filter screen 363 allows the user to slice the OLAP cube using filters on the dimensions. FIG. 5 shows a screenshot of an example filter screen 363 in accordance with an embodiment of the present invention. In the example of FIG. 5, selection areas 502, 504, 506, 508, 510, and 512 are table-driven in that they accept dimensions that are selectable based on information from control tables in the OLAP database 174. For example, once the user selects a dimension for selection area 508, a front end program 220 may consult a dimensions control table 342 to determine the kind of the selected dimension and the conditional operators that may be used for the selected dimension. The front end program 220 may consult a hierarchy control table 344 to determine which hierarchy

table 340 to use and to perform translations between the identifier and actual name of a dimension. Dimensions that serve as possible values for the selected dimension are then displayed by a user interface 360 in selection area 512. In the example of FIG. 5, the user is requesting a report for the dimensions specified in a layout screen 362, but limited to a "category" that is "equal" to a category selected from "Automative," "Business,"...etc. The user is further limiting the report to the "current month" between "10/01/2003" and "10/31/2003." Note that once the user selects a "date range," the selectable values for "start date" and "end date" will be set based on the data available from the OLAP database 174. Similarly, once the user selects "category" in selection area 508, the conditional operators to choose from in selection area 510 and the categories to choose from in selection area 512 are set based on data in the OLAP database 174. As can be appreciated, this advantageously allows addition of more dimensions in the OLAP database 174 without necessarily having to increase the number of selection windows in the user interface 360.

In one embodiment, a front end program 220 has minimal processing load to allow it to adapt to a changing OLAP database 174 and to allow it to be more portable to other databases. Accordingly, in one embodiment, a front end program 220 works in a client-server relationship with an OLAP database 174 and is driven by tables in the OLAP database 174. This advantageously obviates the need for hard coding of available dimensions in the front end program 220 and offloads the processing burden to the OLAP database 174 (which may be running in a relatively fast server computer). For example, the front end program 220 may receive a dimensions control table 342

and display the exhaustive list of dimensions and facts in the dimensions control table 342 in window 410 (see FIG. 4).

As a further example, the values selectable from selection areas 510 and 512 are based on a value selected by a user in selection area 508 (see FIG. 5). The valid values to be displayed in selection areas 510 and 512 may be determined from a dimensions control table 342 and a hierarchy control table 344.

A front end program that offloads the majority of processing to a server computer is especially important in online advertising. Unlike in mail order or catalog advertising applications, advertisement delivery over the Internet is essentially free in that an advertiser can send additional advertisements without incurring substantial additional cost. This results in a relatively large volume of impression data. A front end program 220 advantageously allows a client computer to access large amounts of data by using a back end server to do the "heavy lifting." This will have the desirable effect of scalability by essentially scaling the back end as opposed to being reliant on client computer resources (which in many ways are outside the control of the programmer). This also allows for robust programming and the ability to retrieve data even with relatively old or low capacity client computers.

A report screen 364 provides a report based on the dimensions and facts selected by a user in a layout screen 362 as filtered by the dimensions the user selected in a filter screen 363. A front end program 220 may provide the selected dimensions and facts to a procedure 175 in a message server 163 (see FIG. 2). The procedure 175 may then query an OLAP database 174, filter the result of the query using the dimensions selected in the filter screen 363, and provide the result to the front

end program 220. The result may be viewed in a report screen 364. FIG. 6 shows a screenshot of an example report 600 displayed in a report screen 364 in accordance with an embodiment of the present invention. In the example of FIG. 6, the report screen 364 shows the impressions, clicks, and CTR for the advertiser "Accucard." A user may select a different view of the report by selecting another view in the navigation window 604. The dimensions and facts for a report may be saved as a "view." As will be further explained below, save views may be rerun to take advantage of new data in an OLAP database 174. The dimensions for the report may be further configured by specifying new dimensions. An OLAP database 174 may have to be re-queried if the saved view does not include the data for the new dimensions, or if the user elects to rerun the view.

In the example of FIG. 6, new values for the dimension "ByDay" may be selected from a pull-down menu 602. A report may also be pivoted to show a different view. For example, a report may be pivoted by replacing one dimension with another.

A scheduler 374 may comprise computer-readable program code for scheduling report generation. Scheduler 374 may run reports based on a previously saved view, which comprise dimensions and facts selected by a user in a layout screen 362 and a filter screen 363. A scheduler 374 may be configured to generate a report at a certain frequency, for a certain period of time. A scheduler 374 may also be configured to save the generated reports in a format supported by productivity programs 222 (e.g., saved in Excel format), and email the generated report to the user. The mechanics of converting data formats and emailing are, in general, known in the art and not further described here. FIG. 7 shows a screenshot of an example screen 710 for a scheduler 374.

An alerts 376 may comprise computer-readable program code for alerting a user about changes in the data stored in an OLAP database 174. An alerts 376 may run a report based on dimensions and facts selected by a user in a layout screen 362 as filtered by dimensions the user selected in a filter screen 363. Thereafter, the alerts 376 may compare the generated report to the alert conditions specified by the user. If the result meets the alert conditions, the alerts 376 may so inform the user. FIG. 8 shows a screenshot of an example screen for an alerts 376. In the example of FIG. 8, the user has requested to be alerted in the event the number of impressions for a specified set of dimensions (not shown) have decreased by 10% in a rolling 7-day average. As can be appreciated, an alerts 376 may be employed to automatically watch for trends.

Turning to FIG. 9, there is shown a flow diagram of a method of generating a report in accordance with an embodiment of the present invention. In step 902, a procedure for an OLAP database receives dimensions selected by a user. The dimensions may relate to online advertising. The dimensions may be selected by making selections or entering values in a front end program for the OLAP database.

In step 904, the procedure checks a hierarchy control table for the highest level hierarchy table (referred to as "relevant hierarchy table") containing all of the selected dimensions. In step 906, the procedure may consult a dimensions control table to get the name of the relevant hierarchy table. In step 908, the procedure creates a query to extract dimensions and facts from the hierarchy table. A general algorithm to construct a SQL query is shown in Table 1:

TABLE 1

1) Construct a where clause using the filters set.

a. Convert each of the filter elements into Ids (identifications) using the lookup tables. In the aggregate tables, the filter values are stored as Ids. This allows for a compact storage of long descriptive names. Hence the query string for the OLAP table itself has to be constructed using Ids rather than the descriptive elements that the user sets in the filter.
b. Construct a where clause with the converted filter ID values
    i. Use the OR condition to choose for ID values within each line
    ii. Use the AND condition to choose across different dimensions e.g. (dim1 = value1 or dim1 = value2) and (dim2 = value3 or dim3 = value4)

2) Construct the select statement using the dimensions and facts in the layout
3) Construct the group by statement using the dimensions in the layout
4) Construct the order by statement using the default sort order for each dimension in the layout. The default sort order is stored in the dimension lookup table.
5) Construct the from statement using the table name that contains the highest level of aggregation that contains all the dimensions and facts that are called for in the filters and in the layout.

In step 910, the procedure may enforce filter rules on the result of the query. The filter rules may be based on dimensions selected by the user in a filter screen of the front end program. In one embodiment, the result of a query is filtered by applying a "where condition" and a "select statement." The "where condition" may specify the data to be extracted from the relevant hierarchy table. For example, the pseudo code:

select from "relevant_hierarchy_table";

where advertiser=5 and message_type=7;

extracts facts for the dimension advertiser with a value of "5" ("5" may refer to a specific advertiser, such as Vendor, Inc.) and the dimension message_type with a value of "7" (again, "7" may refer to a specific message type, such as a pop-up) from the relevant hierarchy table.

The procedure may forward the filtered result of the OLAP database query to the front end program, which then formats the result for presentation to the user as a report.

While specific embodiments of the present invention have been provided, it is to be understood that these embodiments are for illustration purposes and not limiting.

Many additional embodiments will be apparent to persons of ordinary skill in the art reading this disclosure.